EDAA75 Exam

EDAA75-2023 Exam #2

2023-08-22

Things you CAN use during the exam:

Any written or printed material is fine. Textbook, other books, the printed slides, handwritten notes, whatever you like.

In any case, it would be good to have a source for the relevant definitions, and also for notation, just in case you don't remember the precise definition of everything we discussed in the course.

Things you CANNOT use during the exam:

Anything electrical or electronic, any communication device: computers, calculators, mobile phones, toasters, …

WRITE CLEARLY. If I cannot read/decipher/make sense of something you write, I will make the least favourable assumption about what you intended to write.

A sheet with common symbols and notations and with information about grading is attached at the end.

Possible solution or hints: This version includes a reference solution, marked like this paragraph. **Note:** the reference solution often offers additional explanations / proof sketches beyond what the question asked for, to help students who use it to study for future exams. Students were not required to explain their answers unless the question explicitly requested an explanation.

Good luck!

Question:	1	2	3	4	5	**Sum**
Max Points:	16	16	16	18	18	84
Points Reached:						

Total points:	100 + lab bonus
Points required for **3**:	50
Points required for **4**:	67
Points required for **5**:	85

Question 1 (16 Points)

In this question, all intervals of the form $[a, b]$ are subsets of $\mathbb{Z}$.

If you encounter unfamiliar notation, make sure to check the notation table on the last page.

(a) Let $S = \{s \cdot n^2 \mid s \in \{-1, 1\}, n \in \mathbb{Z}\}$. Give a function $g : \mathbb{N} \to S$ such that g is injective, but not surjective.

$$g = x \mapsto x^2$$

(b) Which of the following formulas hold? For each row, determine if the formula *always* holds, *never* holds, or is *contingent*. If the formula *always* holds, write *always* in the **Positive** column. If the formula *never* holds, write *never* in the **Positive** column. Otherwise give one example of a truth value assignment (e.g., $a = \mathbf{T}, b = \mathbf{F}$) that ensures that the formula holds (in the **Positive** column) and one example of a variable assignment that ensures that the formula does not hold (in the **Negative** column).

Formula	Positive	Negative
$b \wedge \neg b$	*never*	
$b \vee \neg b$	*always*	
a	$a = \mathbf{T}$	$a = \mathbf{F}$
$(a \vee b) \to \neg a$	$a = \mathbf{F}, b = \mathbf{T}$	$a = \mathbf{T}$
$a \vee \neg(b \to \neg a)$	$a = \mathbf{T}$	$a = \mathbf{F}, b = \mathbf{F}, c = \mathbf{F}$
$\neg((\neg a) \wedge (b \vee \neg a))$	$a = \mathbf{T}, b = \mathbf{T}$	$a = \mathbf{F}, b = \mathbf{T}$
$(a \leftrightarrow b) \wedge (a \leftrightarrow \neg b)$	*never*	

Note on grading: Each row (except for the three example rows) is worth two points, one per column. Omitting a column counts as ± 0. Answering a column incorrectly counts as -1. The answers *always*/*never* count for both columns. A negative points total on this table counts as zero points total for the table.

(c) Are the following formulas ϕ and ψ equivalent for every $S \subseteq \mathbb{Z}$? Explain your answer.

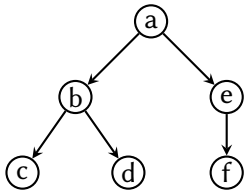
$$\begin{aligned} \phi &= \forall x \in S.\, (\exists y \in S.\, (x \cdot y \in S)) \\ \psi &= \exists y \in S.\, (\forall x \in S.\, (x \cdot y \in S)) \end{aligned}$$

Possible solution or hints: No. Counter-example: $S = \{-2, -1, 2\}$.

- For ϕ, $x \in \{-2, 2\}$ are covered by $y = -1$. With $x = -1$, we can select $y = 2$.
- For ψ, there is no single y that is suitable. $y = -1$ does not work with $x = -1$. $y \in \{-2, 2\}$ do not work with $x = 2$.

Question 2 (16 Points)

In this question we examine and characterise nodes that share ancestors. As an example, consider the following directed tree:



Siblings: {c, d}, {b, e}
First Cousins: {c, f}, {d, f}

Here, c and d are *siblings*, since they have the same parent, namely b. b and e are also siblings, with the same parent, a. Meanwhile, c and f are *first cousins*, since they share the same *grandparent* (parent-of-their-parent), namely a. Nodes d and f are also first cousins. However, a node is never its own sibling or its own first cousin, and two nodes that are siblings are not first cousins.

When we look at arbitrary directed trees (which may be of any size), we can generalise this idea to nth cousins: two nodes are *second cousins* when they share a great-grandparent, *third cousins* when they share a great-great-grandparent, and so on. In this format, a "0th cousin" is a sibling. As before, no node is its own nth cousin, and two nodes cannot be nth cousins if they are already mth cousins and $m < n$.

For an arbitrary (directed) tree $\langle T, R \rangle$, answer the following sub-questions. You may re-use definitions from *earlier* sub-questions (only), even if you have not solved those sub-questions.

(a) Define three relations $I \subseteq T \times T$, $S \subseteq T \times T$, and $Q \subseteq T \times T$, where xIy iff x and y are the same element, xSy whenever x and y have the same parent, and xQy whenever x and y have the same grandparent (= parent of their own parent).

$$I = \{\langle t, t \rangle \mid t \in T\}$$

$$S = R \circ R^{-1}$$

$$Q = R \circ R \circ R^{-1} \circ R^{-1}$$

(b) Now generalise S and Q. *Recursively* define a *family of relations* $P_n \subseteq T \times T$ with $n \in \mathbb{N}$ such that for all $n \geq 1$, the following holds: xP_ny iff x and y share some ancestor z, there exists a path from z to x of length n, and there exists a path from z to y of length n. In particular, $P_1 = S$ and $P_2 = Q$.
(Hint: you can choose P_0 freely, and it won't affect your grade. You may want to pick something that makes your life easy.)

$$P_n = \begin{cases} I & \text{if } n = 0 \\ R \circ P_{n-1} \circ R^{-1} & \text{if } n > 0 \end{cases}$$

(c) Why is your recursive definition well-founded (i.e., why does it not loop infinitely)? Explain in your own words.

Possible solution or hints: Each P_{n+1} for $n \in \mathbb{N}$ references P_n, and P_0 does not reference any P_i for any $i \in \mathbb{N}$. Hence, each P_{n+1} will have precisely n recursion steps, as we can see by induction.

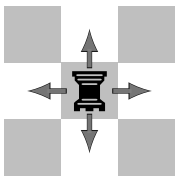
(d) Define another family of relations C_n with $n \in \mathbb{N}$ such that xC_ny iff x and y are nth cousins. (E.g., C_0 should describe the sibling relation.) Do *not* define C_n recursively. You may re-use any existing recursive definitions.

$$C_n = P_{n+1} \setminus P_n$$

Question 3 (16 Points)

In this question we develop a chess-like game that we call VChess, short for *variant chess*. You don't need to know anything about chess to answer the sub-questions below.

A VChess game involves different game pieces p that can move across the tiles of a game board. We can describe the game pieces' possible moves as vectors in two-dimensional Euclidean space. For example, the game piece below (♜) can move exactly one tile up, down, left, or right, and we can represent each of these movements as an element $\langle d_x, d_y \rangle \in \mathbb{Z}^2$, where d_x is the horizontal (left-right) movement and d_y is the vertical (up-down) movement:



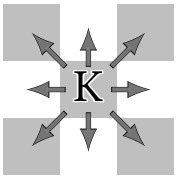
Move	d_x	d_y
Left	$\langle -1,$	$0\rangle$
Right	$\langle\ 1,$	$0\rangle$
Down	$\langle\ 0,$	$-1\rangle$
Up	$\langle\ 0,$	$1\rangle$

We call the set of all possible moves of some game piece p the *move set* of p, $M_p \subseteq \mathbb{Z}^2$. For example, a piece r that can move diagonally left-and-down has $\langle -1, -1 \rangle \in M_r$, and a piece q that can move arbitrarily far up has $\langle 0, n+1 \rangle \in M_q$ for all $n \in \mathbb{N}$. For our example above, we have:

$$M_{♜} = \{\langle -1, 0\rangle, \langle 1, 0\rangle, \langle 0, -1\rangle, \langle 0, 1\rangle\}$$

Solve the sub-questions below without giving any recursive definitions. Your definitions below may re-use your own definitions from earlier sub-questions (only).

(a) (Warm-up question) The game piece **K** can make the same moves as a ♜, or alternatively take one step in any of the four diagonal directions:



Give the move set of a **K**.

$$M_{\mathbf{K}} = M_{♜} \cup \{\langle 1, 1\rangle, \langle -1, 1\rangle, \langle 1, -1\rangle, \langle -1, -1\rangle\}$$

(b) Most game pieces' move sets follow a form of *rotational symmetry*: if the game piece can move to the right ($\langle 1, 0\rangle \in M_p$), then the game piece can also move down ($\langle 0, -1\rangle \in M_p$), or up, or to the left. Similarly, if the game piece can move diagonally up-and-left, it can also move diagonally in the other three directions.

Give a function $r_{90} : \mathbb{Z}^2 \to \mathbb{Z}^2$ that rotates a move by 90 degrees (clockwise or anticlockwise, your choice. You do *not* need trigonometric functions like $\sin$ and $\cos$ here.).

$$r_{90} = \langle d_x, d_y\rangle \mapsto \langle -d_y, d_x\rangle \text{ or } \langle d_y, -d_x\rangle$$

(c) Describe this rotational symmetry by giving a relation R with the property that xRy iff x is rotationally symmetric to y, by which we mean that calling r_{90} repeatedly on x yields y (i.e., $y = r_{90}(r_{90}(\ldots r_{90}(x)\ldots))$).

$$R = \{\langle a, b\rangle \mid a \in \mathbb{Z}^2, b \in \mathbb{Z}^2, b \in \{ a, r_{90}(a), (r_{90} \circ r_{90})(a), (r_{90} \circ r_{90} \circ r_{90})(a)\} \}$$

Possible solution or hints: Note that $r_{90} \circ r_{90} \circ r_{90} \circ r_{90}$ is the identity function $x \mapsto x$ (rotating by 360 degrees), so we don't need to consider rotating by 360 degrees, by 450 degrees etc.

(d) Construct a *rotational closure* function $c_R : \mathcal{P}(\mathbb{Z}^2) \to \mathcal{P}(\mathbb{Z}^2)$ such that $c_R(M)$ is the smallest superset of M ($c_R(M) \supseteq M$) such that $m \in c_R(M)$ iff for all $m' \in \mathbb{Z}^2$ with mRm', we have that $m' \in c_R(M)$ holds.

$$c_R = M \mapsto R(M)$$

Question 4 (18 Points)

A small company uses four-digit codes to control access to their systems, and each employee can pick their own code. After a break-in caused by someone using code `0000`, the company is tightening security and considers the following policies:

- **A**: No digit may occur more than once per access code.
- **B**: No digit may be followed by a digit that is exactly one higher or one lower than the previous digit, and the sequences `09` and `90` may not appear anywhere in an access code.

When answering the questions below, you do not need to evaluate exponentials, factorials, products, sums, quotients, or subtractions.

(a) How many possible access codes are there without these policies?

Possible solution or hints: 10^4 (= 10000)

(b) How many possible access codes would there be with policy **A**?

Possible solution or hints: $\frac{10!}{(10-4)!}$ (= 5040)

(c) How many possible access codes would there be with policy **B**?

Possible solution or hints: $10 \cdot 8^3$ (= 5420). Explanation: first digit is free to choose, the next digits are always constrained to a choice of eight.

(d) The company has decided to make the following changes:

1. All access codes will now be made up of **three symbols** instead of four digits.
2. The set of symbols is $S = \{s_1, \ldots, s_n\}$, but we do not yet know how many symbols $\#S = n$ there will be. Several options are still being considered, including the letters of the Latin alphabet ($n = 26$, $s_1 = \texttt{A}, \ldots, s_{26} = \texttt{Z}$), For comparison, the decimal digits would have the form $n = 10$, $s_1 = \texttt{0}, \ldots, s_{10} = \texttt{9}$.
3. Policy **A** is modified, as follows: "No symbol may occur more than once per access code."
4. Policy **B** is modified, as follows: "No symbol s_i may be followed by symbol s_{i-1} (if $i > 1$) or s_{i+1} (if $i < n$). Symbol s_1 must not follow s_n, and symbol s_n must not follow s_1."

How many possible access codes will there be if *both* Policy **A** *and* **B** are implemented, for $n \in \mathbb{N}$ and $n > 10$? Explain your answer. Focus on explaining your answer; you don't need to simplify the arithmetic term that you give. [**NB:** The exam as originally given asked a slightly different question; cf. the reference solution. The question above has been updated to make this exam more useful for studying / pre-exam practice.]

Possible solution or hints: The exam originally asked "How many possible access codes will there be now, for $n \in \mathbb{N}$ and $n > 10$? Explain your answer." Thus, it was sufficient to answer separately for Policy **A** and **B**. Students who answered for both policies together got an extra +4 bonus points.

Answer for *no policy (this was entirely optional)*:

$$\#C = n^3$$

where $C = S^3$ is the set of all three-symbol codes.

Answer for *only Policy **A***:

$$\#C_A = \frac{n!}{(n-3)!}$$

where C_A is the set of all three-symbol codes that follow the modified Policy **A**. As in subquestion (c), the boundary cases simply "wrap around", analogously to the "modulo" operation: for $x = 1$, we simply replace s_0 by s_n, and for $x = n$, we replace s_{x+1} by s_1. Thus, we don't need to treat the cases $x = 1$ and $x = n$ any differently from the cases where $1 < x < n$.

Answer for *only Policy **B***:

$$\#C_B = n \cdot (n-2)^2$$

where C_B is the set of all three-symbol codes that follow the modified Policy **B**.

(Everything below is only relevant for the "bonus points" variant of the question.)

Answer for *both Policy **A** and Policy **B** together*: We are looking for $\#(C_A \cap C_B)$.

There are several approaches to this problem;

Approach 1: Perhaps the easiest approach (if we spot it) is to work out that for all codes $s_x s_y s_z$, we can see that s_y is restricted to $n - 3$ choices (not in $\{s_x, s_{x-1}, s_{x+1}\}$), and z is restricted to $n - 4$ choices, no matter what choice we make for y: since $y + 1 \neq x$ and $y - 1 \neq x$, the number of choices that we are not allowed to take for y is always 4. Thus, we have the following number of choices:

$$n \cdot (n-3) \cdot (n-4)$$

Approach 2: Instead of working with the set of all codes that are allowed, we can also try to work out the set of all codes that are *forbidden* by both. Let $\overline{C_A} = C \setminus C_A$ and

$\overline{C_B} = C \setminus C_B$ (SLAM also writes this as $-C_A$ and $-C_B$, if the "universe" U is set to C, cf. SLAM, page 23). In this approach, we then want to transform $\#(C_A \cap C_B)$ into a term whose only unknown is $\#(\overline{C_A} \cap \overline{C_B})$ This approach ends up being a bit more complicated, but will include it here, since it is a perfectly fine approach to try and demonstrates several useful principles.

First, we exploit that $C_A \cap C_B = C \setminus (\overline{C_A} \cup \overline{C_B})$ (cf. SLAM exercise 1.4.3(e); we are applying DeMorgan's rules here). The term that we want to keep as the unknown, $\#(\overline{C_A} \cap \overline{C_B})$, will then pop up as soon as we apply the Addition Principle. Specifically:

$$
\begin{aligned}
\#(C_A \cap C_B) &= \#(C \setminus ((C \setminus C_A) \cup (C \setminus C_B))) \\
&= \#(C \setminus (\overline{C_A} \cup \overline{C_B})) \\
&= \#C - \#(C \cap (\overline{C_A} \cup \overline{C_B})) && \text{(Subtraction Principle)} \\
&= \#C - \#(\overline{C_A} \cup \overline{C_B}) && (\overline{C_A} \cup \overline{C_B} \subseteq C) \\
&= \#C - (\#\overline{C_A} + \#\overline{C_B} - \#(\overline{C_A} \cap \overline{C_B})) && \text{(Addition Principle)} \\
&= \#C - \#\overline{C_A} \qquad\quad -\#\overline{C_B} + \#(\overline{C_A} \cap \overline{C_B}) \\
&= \#C - \#(C \setminus C_A) \qquad -\#\overline{C_B} + \#(\overline{C_A} \cap \overline{C_B}) \\
&= \#C - (\#C - \#(C \cap C_A)) - \#\overline{C_B} + \#(\overline{C_A} \cap \overline{C_B}) && \text{(Subtraction Principle)} \\
&= \#C - (\#C - \#C_A) \qquad -\#\overline{C_B} + \#(\overline{C_A} \cap \overline{C_B}) && (C_A \subseteq C) \\
&= \#C - (\#C - \#C_A) - (\#C - \#C_B) + \#(\overline{C_A} \cap \overline{C_B}) && \text{(Subtraction principle)} \\
&= \#C - \#C + \#C_A - (\#C - \#C_B) + \#(\overline{C_A} \cap \overline{C_B}) \\
&= \#C_A - (\#C - \#C_B) + \#(\overline{C_A} \cap \overline{C_B}) \\
&= \#C_A - \#C + \#C_B + \#(\overline{C_A} \cap \overline{C_B})
\end{aligned}
$$

As we intended, the only unknown term in this sum is $\#(\overline{C_A} \cap \overline{C_B})$, since we already know:

$$
\begin{aligned}
\#C_A &= \tfrac{n!}{(n-3)!} \\
\#C &= n^3 \\
\#C_B &= n \cdot (n-2)^2
\end{aligned}
$$

To find $\#(\overline{C_A} \cap \overline{C_B})$, observe that policy **B** allows repetition but disallows symbols in sequence to go up or down by one, so any code that violates both **B** and **A** must both contain a sequence $s_x, s_x + 1$ or s_x, s_{x-1} (modulo the boundary cases $x = 1$ and $x = n$) and also have x or $x - 1$ repeated before or after that sequence.

We can work out the number of unique sequences for any given $x \in [1, n]$ using the Multiplication Principle: s_x must appear twice, and either s_{x+1} or s_{x-1} must appear once, in any position. So we have two choices (s_{x+1} or s_{x-1}) of *what* to insert, and three choices of *where* to insert (first, middle, or last symbol), which gives $2 \cdot 3 = 6$ choices per x, so with n choices for x, we have $6 \cdot n$ codes that are forbidden by both Policy **A** and Policy **B**.

We then arrive at the following solution:

$$\#(C_A \cap C_B) = \frac{n!}{(n-3)!} - (n^3 - n \cdot (n-2)^2) + (6 \cdot n)$$

Alternatively (or if the above derivation of $6 \cdot n$ seems a bit too fast), we can list all of these possible combinations:

- $c_1 = s_y, s_x, s_{x-1}$, where $y = x$ or $y = x - 1$
- $c_2 = s_y, s_x, s_{x+1}$, where $y = x$ or $y = x + 1$
- $c_3 = s_x, s_{x-1}, s_y$, where $y = x$ or $y = x - 1$
- $c_4 = s_x, s_{x+1}, s_y$, where $y = x$ or $y = x + 1$
- $c_5 = s_{x-1}, s_y, s_x$, where $y = x$ or $y = x - 1$

- $c_6 = s_{x+1}, s_y, s_x$, where $y = x$ or $y = x + 1$

This gives us 12 different sequences. However, there is a pitfall here: while all of the above sequences are indeed codes that both violate Policy **A** and Policy **B**, they are not unique, so we don't actually get $12 \cdot n$ sequences in total. As an example, set $n = 10$. Then,

- c_1 with $x = 5$ and $y = x - 1$ would give the code $s_4 s_5 s_4$. However, c_4 with $x = 4$ and $y = x$ gives us the same code.
- c_2 with $x = 5$ and $y = x$ would give the code $s_5 s_5 s_6$. However, c_5 with $x = 6$ and $y = x - 1$ gives us the same code.
- c_3 with $x = 5$ and $y = x - 1$ would give the code $s_5 s_4 s_4$. However, c_6 with $x = 4$ and $y = x$ gives us the same code.

If we check all of $c_1, \ldots, c_6$, we find that we are counting every combination twice!
To avoid this ambiguity, we can limit ourselves, for a given x, to only those combinations in which s_x appears exactly twice. That is analogous to fixing $y = x$. That leaves us at only one possible way to construct $c_1, \ldots, c_6$ for any given x and brings us back to $6 \cdot n$.

To see that the two solutions are identical: While this goes beyond the exercise, we can show that our who solutions for $\#(C_A \cap C_B)$ are identical:

$$
\begin{aligned}
\tfrac{n!}{(n-3)!} - (n^3 - n(n-2)^2) + 6n &= \tfrac{n!}{(n-3)!} - n(n^2 - (n-2)^2) + 6n \\
&= \tfrac{n!}{(n-3)!} - n(n^2 - (n^2 - 4n + 4)) + 6n \\
&= \tfrac{n!}{(n-3)!} - n(n^2 - n^2 + 4n - 4) + 6n \\
&= \tfrac{n!}{(n-3)!} - n(4n - 4) + 6n \\
&= \tfrac{n!}{(n-3)!} - n(4n - 4 - 6) \\
&= \tfrac{n!}{(n-3)!} - n(4n - 10) \\
&= n \cdot (n-1) \cdot (n-2) - n(4n - 10) \\
&= n((n-1) \cdot (n-2) - (4n - 10)) \\
&= n(n^2 - 3n + 2 - 4n + 10) \\
&= n(n^2 - 7n + 12) \\
&= n(n-3)(n-4)
\end{aligned}
$$

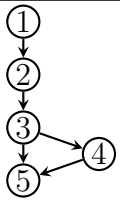
Question 5 (18 Points)

In this question we analyse software with the help of graphs. A *guarded graph* is a tuple $\langle G, V, C, g, a\rangle$ where $\langle V, E\rangle$ is a graph of program statements V, with xEy stating that y can execute directly after x. Some edges $e \in E$ represent 'if' statements, which we model with *conditions* $C \supseteq \{\mathbf{T}\}$ and functions $g : E \to C$ and $a : V \to C$. We say that an edge $e \in E$ is *unguarded* iff $g(e) = \mathbf{T}$, and that e is *guarded* by $g(e)$ iff $g(e) \neq \mathbf{T}$. We say that vertex $v \in V$ *activates* condition c iff $a(v) = c$.

For a guarded graph $\langle V, E, C, g, a\rangle$, a *guarded path* of *length* $n-1$ from v_1 to v_n is a path $v_1, \ldots, v_n$ in $\langle V, E\rangle$, where additionally for all $j \in [1, n)$, $g(\langle v_j, v_{j+1}\rangle) = c$ implies that either $c = \mathbf{T}$, or there exists some $k \in \mathbb{N}, k \leq j$ such that $a(v_k) = c$.

As an example, the program below has $C = \{\mathsf{b}, \mathbf{T}\}$, and $\langle V, E\rangle$, g, and a as follows:

Program (for illustration only)	**Graph** $\langle V, E\rangle$	$g : E \to C$	$a : V \to C$

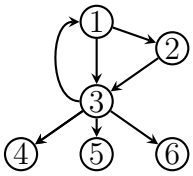
`1 start();`
`2 b = true;`
`3 if (b) {`
`4  secret();`
`5 }`
		$g(\langle 1,2\rangle) = \mathbf{T}$
		$g(\langle 2,3\rangle) = \mathbf{T}$
		$g(\langle 3,4\rangle) = \mathsf{b}$
		$g(\langle 3,5\rangle) = \mathbf{T}$
		$g(\langle 4,5\rangle) = \mathbf{T}$
			$a(1) = \mathbf{T}$
			$a(2) = \mathsf{b}$
			$a(3) = \mathbf{T}$
			$a(4) = \mathbf{T}$
			$a(5) = \mathbf{T}$

Here, the sequence 1, 2, 3, 4, 5 is a guarded path because all adjacent vertices are edges, and because the only guarded edge, $\langle 3, 4\rangle$, is guarded by b, which vertex 2 activates (i.e., $a(2) = \mathsf{b}$).

However, $3, 4, 5$ is *not* a guarded path, because $\langle 3, 4\rangle$ is guarded by b, vertex 3 does not activate b, and there is no other vertex before 3 in this path that could activate b.

Answer the sub-questions below.

(a) Consider the following guarded graph with $C = \{\mathbf{T}, \mathsf{c}, \mathsf{d}, \mathsf{e}\}$:

Graph $\langle V, E\rangle$	$g : E \to C$	$a : V \to C$
	$g(\langle 1,2\rangle) = \mathsf{d}$ $g(\langle 1,3\rangle) = \mathbf{T}$ $g(\langle 2,3\rangle) = \mathbf{T}$ $g(\langle 3,1\rangle) = \mathbf{T}$ $g(\langle 3,4\rangle) = \mathsf{e}$ $g(\langle 3,5\rangle) = \mathsf{c}$ $g(\langle 3,6\rangle) = \mathbf{T}$	$a(1) = \mathbf{T}$ $a(2) = \mathsf{c}$ $a(3) = \mathsf{d}$ $a(4) = \mathsf{e}$ $a(5) = \mathbf{T}$ $a(6) = \mathbf{T}$

List all vertices that can be the last vertex of a guarded path that is of length n and starts at vertex 1.

[**NB**: in the exam as given, this was formulated as "List all vertices that you can reach with a guarded path of length n, starting at vertex 1"]

Length	**Reachable vertices**
$n = 1$:	{ 3 } (with d)
$n = 2$:	{6, 1 (with d)}
$n = 3$:	{2 (with c, d), 3 (with d)
$n = 4$:	{3 (with c, d), 6 (with d), 1 (with d)
$n = 5$:	{1,3 (with d), 2,5,6 (with c, d) }

Possible solution or hints: The above (minus the "with ... bits") is sufficient as a solution. Since "reachable" (in the version of the original exam) does not specifically only talk about the final node in the path, it was also correct to list all nodes traversed by the path, as long as the interpretation of "reachable" was consistent, possibly including 1. Under this alternative interpretation, the set of reachable vertices grows monotonically, with $n = 0$ implicitly being either $\emptyset$ or $\{1\}$. Then, $n = 1$ adds 3, $n = 2$ adds 6 and possibly 1, $n = 3$ adds 2, $n = 4$ does not add anything, and $n = 5$ adds 4 to the previous set.

(b) We analyse guarded graphs $\langle E, V, C, g, a\rangle$ with the help of *abstract program states*:
Let $S = \{\langle v, A\rangle \mid v \in V, \{\mathbf{T}, a(v)\} \subseteq A \subseteq C\}$. Then any $s \in S$ with $s = \langle v, A\rangle$ is an *abstract program state* at vertex v, and s *satisfies* condition $c \in C$ iff $c \in A$.

Define a *successor* relation $N \subseteq S \times S$ such that $s_1 N s_2$ if and only if s_1 is at vertex v_1 and s_2 is at vertex v_2, and there is an edge from v_1 to v_2 that is either unguarded or guarded by some condition c such that s_1 satisfies c.

$$
\begin{aligned}
N = \{\ & \langle\langle v, A\rangle, \langle v', A'\rangle\rangle \in S \times S \\
& \mid \langle v, v'\rangle \in E \\
& \wedge g(\langle v, v'\rangle) \in A \\
& \wedge \{\mathbf{T}\} \subseteq A \\
& \wedge A' = A \cup \{a(v)\}\}
\end{aligned}
$$

(c) Is N from sub-question (b) *always*, *never*, or *sometimes* a function? Explain your answer.

Possible solution or hints: Sometimes: if each graph node has precisely one successor and all edges are unguarded, then there is exactly one successor vertex for every vertex and hence $N(s)$ exists and is is unique for any s. However, N is not a function in our earlier examples.

(d) Use N from sub-question (b) to give a logical formula or a term in set theory that holds if and only if there is a guarded path from $v_s \in V$ to $v_e \in V$ for some $\langle E, V, C, g, a\rangle$.

Possible solution or hints: $\exists A \subseteq C.\langle v_s, \{\mathbf{T}\}\rangle N^+ \langle v_e, A\rangle$, which excludes paths of length 0, as in class. Reflexive+transitive closure (N^*) was also acceptable for full credit.

Symbols and Notation

You may use any of the symbols and notation below in your own answers, in addition to any standard arithmetic notation and notation that we discussed in class. You may at any time introduce helper definitions.

$\mathbb{Z}$	The integers
$\mathbb{R}$	The real numbers
$\mathbb{N}$	The natural numbers, starting at 0
$\mathcal{P}(A)$	The power set of the set A
$\#S, \lvert S\rvert$	The cardinality of set S
$\mathsf{dom}(R), \mathsf{dom}(f)$	The domain of a binary relation R or a function f
$\mathsf{range}(R), \mathsf{range}(f)$	The range of a binary relation R or a function f
R^{-1}, f^{-1}	The inverse of a relation R or a function f
$R \circ S, f \circ g$	The composition of relations and the composition of functions
R^+	The transitive closure of relation R
$R[X], f[X]$	closure of a set X under a relation R, a set of relations R, or a function f
$[a, b]$	closed interval from a to b (including $\{a, b\}$)
(a, b)	open interval from a to b (excluding $\{a, b\}$)
$(a, b], [a, b)$	half-open intervals from a to b
$\lfloor x \rfloor$	rounding down x
$\sum S$	sum of all elements of S
$\prod S$	product of all elements of S
$\bigcup S$	union of all elements of S
$\bigcap S$	intersection of all elements of S
$\bigcup_{a \in S} E(a), \bigcap_{a \in S} E(a)$	generalised union / intersection of all sets $E(a)$ for every $a \in S$
$\binom{n}{k}$	Binomial coefficients: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$

Answers to common questions about grading:

- In any question, if sub-question x refers to sub-question y with $x \neq y$, then grading for sub-question x assumes that you answered sub-question y correctly, even if you did not. However, what the correct answer for sub-question x is may depend on your answer to sub-question y.